

Examen

Date : 26/05/2024

Durée : 1h30

Questions (05 pts) : Répondre brièvement aux questions suivantes :

1. Dans quels cas est-il préférable d'utiliser des éléments plutôt que des attributs dans la conception d'un document XML (02 raisons) ?
2. Citer quelques avantages de Schéma XML par rapport la DTD (plus de trois avantages)
3. Pourquoi CSS ne peut pas remplacer XSL dans la mise en forme des document XML ? A votre avis comment peut-on contourner ces limitations ?

Exercice 01 (3+5+3+2+2):

On décrit un ensemble d'animaux de manière simplifiée sous le format suivant :

```
<animaux auteur="Matthias Colin" date="2011-06-27">
  <mammifere nom="girafe" id="a1" locomotion="marche" />
  <mammifere nom="ours" id="a2" locomotion="marche" >
    <mange ref="a3" />
  </mammifere>
  <poisson nom="saumon" id="a3" locomotion="nage" />
  <mammifere nom="dauphin" id="a4" locomotion="nage" />
  <oiseau nom="aigle royal" id="a5" locomotion="vol" >
    <mange ref="a7" />
    <mange ref="a8" />
  </oiseau>
  <reptile nom="crocodile" id="a6" locomotion="nage"/>
  <reptile nom="couleuvre" id="a7" locomotion="reptation"/>
  <mammifere nom="marmotte" id="a8" locomotion="marche" />
  <mammifere nom="lion" locomotion="marche" >
    <mange ref="a1" />
  </mammifere>
</animaux>
```

Cet exemple cite **toutes** les catégories d'animaux et moyens de locomotion prévus.

1. Ecrire une DTD **animaux.dtd** qui décrit un document tel que celui proposé.
2. Ecrire un schéma XSD **animaux.xsd** qui décrit un document tel que celui proposé. On vérifiera que :
 - a) Les noms d'animaux et d'auteur ne contiennent que des caractères alphabétiques et espaces (on ne tient pas compte des caractères accentués, tirets ou apostrophes).

b) La date de la description correspond à un projet scolaire qui se déroule entre le 4 mai 2011 et le 6 juillet 2011.

3. Ecrire une template XSL permettant d'écrire la liste des animaux dans un tableau HTML selon le modèle suivant :

Animal	Famille	Locomotion	Nourriture
girafe	mammifere	marche	
ours	mammifere	marche	saumon
dauphin	mammifere	nage	
marmotte	mammifere	marche	
lion	mammifere	marche	girafe
aigle royal	oiseau	vol	couleuvre, marmotte
saumon	poisson	nage	
crocodile	reptile	nage	
couleuvre	reptile	reptation	

4. Donnez une requête XQuery permettant de construire un document XML suivant la DTD qui suit :

```
<!ELEMENT mammiferes (name)+>
<!ELEMENT name (#PCDATA)>
```

5. Transformer le document XML suivant en document json équivalent :

```
<animaux>
  <mammifere>      <nom>girafe</nom> </mammifere>
  <mammifere>
    <nom> ours </nom>
    <mange> saumon </mange>
  </mammifere>
  <mammifere> <nom> dauphin </nom> </mammifere>
  <mammifere>
    <nom> lion </nom>
    <mange>girafe</mange>
  </mammifere>
  <poisson> <nom> saumon </nom> </poisson>
  <oiseau>
    <nom> aigle royal </nom>
    <mange>couleuvre </mange>
    <mange>marmotte </mange>
  </oiseau>
  <reptile>      <nom> crocodile </nom> </reptile>
</animaux>
```

Corrigé type

Questions (04 pts):

1. Il préférable d'utiliser des éléments plutôt que des attributs dans la conception d'un document XML parce que **(01 pts)**
 - les attributs ne peuvent pas contenir plusieurs valeurs (les éléments le peuvent).
 - les attributs ne peuvent pas contenir de structures arborescentes (les éléments le peuvent).
 - les attributs ne sont pas facilement extensibles (pour les changements futurs).
2. Avantages de Schéma XML par rapport la DTD : **(02 pts)**
 - Prise en charge des espaces de noms
 - Plus de types de données.
 - Capable de créer un type de données complexe.
 - Héritage par extension ou restriction.
3. CSS ne peut pas remplacer XSL dans la mise en forme des document XML parce que CSS fournit un formatage « statique ». Le formatage est basé sur les balises/attributs, pas sur l'organisation.

On peut contourner ces limitations par l'utilisation de javascript/DOM **(0.5+0.5 pts)**

Exercice 01 (16 pts):

1. Ecrire une DTD **animaux.dtd** qui décrit un document tel que celui proposé **(03 pts)**.

```
<!ENTITY % locomotion_type "(marche|nage|vol|reptation)">
<!ELEMENT animaux (mammifere|poisson|oiseau|reptile)+>
<!ATTLIST animaux  auteur CDATA #REQUIRED
                  date CDATA #REQUIRED >
<!ELEMENT mammifere (mange*)>
<!ATTLIST mammifere  id ID #IMPLIED
                  nom CDATA #REQUIRED
                  locomotion %locomotion_type; #REQUIRED>
<!ELEMENT poisson (mange*)>
<!ATTLIST poisson  id ID #REQUIRED
                  nom CDATA #REQUIRED
```

```
                  locomotion %locomotion_type; #REQUIRED>
<!ELEMENT oiseau (mange*)>
<!ATTLIST oiseau  id ID #REQUIRED
                  nom CDATA #REQUIRED
                  locomotion %locomotion_type; #REQUIRED>
<!ELEMENT reptile (mange*)>
<!ATTLIST reptile  id ID #REQUIRED
                  nom CDATA #REQUIRED
                  locomotion %locomotion_type; #REQUIRED>
<!ELEMENT mange EMPTY>
<!ATTLIST mange  ref IDREF #REQUIRED>
```

2. Ecrire un schéma XSD **animaux.xsd** qui décrit un document tel que celui proposé. On vérifiera que **(05 pts)**:
 - c) Les noms d'animaux et d'auteur ne contiennent que des caractères alphabétiques et espaces (on ne tient pas compte des caractères accentués, tirets ou apostrophes).
 - d) La date de la description correspond à un projet scolaire qui se déroule entre le 4 mai 2011 et le 6 juillet 2011.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:simpleType name=" nomtype">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Za-z\s]+" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="locomototype">
  <xs:restriction base="xs:string">
    <xs:enumeration value="marche" />
    <xs:enumeration value="nage" />
    <xs:enumeration value="vol" />
    <xs:enumeration value="reptation" />
  </xs:restriction>
</xs:simpleType>

<xs:element name="animaux">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element name="mammifere" type="animalType" />
      <xs:element name="poisson" type="animalType" />
```

```

<xs:element name="oiseau" type="animalType" />
<xs:element name="reptile" type="animalType" />
</xs:choice>
  <xs:attribute name="auteur" type="nomtype" use="required" />
  <xs:attribute name="date" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:date">
        <xs:minInclusive value="2011-05-04"/>
        <xs:maxInclusive value="2011-07-06"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
<xs:complexType name="animalType">
  <xs:sequence>
    <xs:element name="mange" minOccurs="0"
      maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="ref" type="xs:IDREF"
          use="required" />
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required" />
  <xs:attribute name="nom" type="nomtype" use="required" />
  <xs:attribute name="locomotion" type="locomotype"
    use="required" />
</xs:complexType>
</xs:schema>

```

3. Ecrire une template XSL permettant d'écrire la liste des animaux dans un tableau HTML(04 pts)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8"/>
  <xsl:template match="animaux">
    <html>
      <head>
        <title>Liste des animaux</title>
      </head>
      <body>
        <h1>Liste des animaux</h1>

```

```

<table border="1">
  <tr>
    <th>Animal</th>
    <th>Famille</th>
    <th>Locomotion</th>
    <th>Nourriture</th>
  </tr>
  <xsl:apply-templates select="mammifere" />
<xsl:apply-templates select="oiseau" />
<xsl:apply-templates select="poisson" />
<xsl:apply-templates select="reptile" />
</table>
</body>
</html>
</xsl:template>
<xsl:template match="mammifere">
  <tr>
    <td><xsl:value-of select="@nom" /></td>
    <td><xsl:value-of select="local-name()" /></td>
    <td><xsl:value-of select="@locomotion" /></td>
    <td>
      <xsl:for-each select="mange">
        <xsl:variable name="name" select="@ref"/>
        <xsl:value-of
select="//animaux/*[@id=$name]/@nom"/>
        <xsl:if test="position() != last()",
</xsl:if>
      </xsl:for-each>
    </td>
  </tr>
</xsl:template>
<xsl:template match="oiseau">
  <tr>
    <td><xsl:value-of select="@nom" /></td>
    <td><xsl:value-of select="local-name()" /></td>
    <td><xsl:value-of select="@locomotion" /></td>
    <td>
      <xsl:for-each select="mange">
        <xsl:variable name="name" select="@ref"/>
        <xsl:value-of
select="//animaux/*[@id=$name]/@nom"/>
        <xsl:if test="position() != last()",
</xsl:if>
      </xsl:for-each>
    </td>
  </tr>

```

```

        </xsl:for-each>
    </td>
</tr>
</xsl:template>
<xsl:template match="poisson">
    <tr>
        <td><xsl:value-of select="@nom" /></td>
        <td><xsl:value-of select="local-name()" /></td>
        <td><xsl:value-of select="@locomotion" /></td>
        <td>
            <xsl:for-each select="mange">
                <xsl:variable name="name" select="@ref"/>
                <xsl:value-of
select="//animaux/*[@id=$name]/@nom"/>
                <xsl:if test="position() != last()",
</xsl:if>
            </xsl:for-each>
        </td>
    </tr>
</xsl:template>
<xsl:template match="reptile">
    <tr>
        <td><xsl:value-of select="@nom" /></td>
        <td><xsl:value-of select="local-name()" /></td>
        <td><xsl:value-of select="@locomotion" /></td>
        <td>
            <xsl:for-each select="mange">
                <xsl:variable name="name" select="@ref"/>
                <xsl:value-of
select="//animaux/*[@id=$name]/@nom"/>
                <xsl:if test="position() != last()",
</xsl:if>
            </xsl:for-each>
        </td>
    </tr>
</xsl:template>
</xsl:stylesheet>

```

4. Donnez une requête XQuery permettant de construire un document XML suivant la DTD qui suit :

```

<!ELEMENT mammiferes (name)+>
<!ELEMENT name (#PCDATA)>(02 pts)

```

5. Transformer le document XML suivant en document json équivalent **(02 pts)**:

```

{
  "animaux": {
    "mammifere": [
      {
        "nom": "girafe"
      },
      {
        "nom": "ours",
        "mange": "saumon"
      },
      {
        "nom": "dauphin"
      },
      {
        "nom": "lion",
        "mange": "girafe"
      }
    ],
    "poisson": [
      {
        "nom": "saumon"
      }
    ],
    "oiseau": [
      {
        "nom": "aigle royal",
        "mange": [
          "couleuvre",
          "marmotte"
        ]
      }
    ]
  },
  "reptile": [
    {
      "nom": "crocodile"
    }
  ]
}

```